Shmoo Plotting: The Black Art of IC Testing

KEITH BAKER
Philips ED&T
JOS VAN BEERS
Philips Research Laboratories

Obtaining higher yields from IC fabrication is a never-ending goal. Toward that end, shmoo plotting can help bridge the gap between design and test and ultimately show ways to improve a product, process, or manufacturing test program.

THE DRIVE TO BUILD BETTER, less expensive products is nearly universal, and it is certainly true in the semiconductor industry. IC vendors fabricated early MOS integrated circuits with PMOS technologies as an alternative to compete with established bipolar circuits. Complex and expensive, bipolar circuits nonetheless exhibited superior performance and were less sensitive to external parameters, such as voltage and temperature. Although inexpensive because it uses relatively inexpensive processes, PMOS suffered unstable MOSFET threshold voltage and poor electron mobility, compared to the later CMOS processes.

Checking the performance of an IC's MOSFETs against the bias voltages was crucial. To successfully compete with bipolar ICs, designers needed to find a way to maximize MOS IC performance and stability in the manufacturing process, given the IC's fundamental MOSFET characteristics. These characteristics fundamentally depend on process parameters, temperature, and signal conditions, specifically the MOSFET's transductance and the device threshold voltage.

Shmoo plotting specifically analyzes the performance of a digital IC compared to the major analog parameters that influence the characteristics of the basic building block, the MOSFET. Shmoo plotting offers you a way to visualize the relationship between the performance of an IC and changes in the external environment, such as temperature, V_{DD}, and timing. When used with batches of ICs from process lots, shmoo plotting also allows the influence of production process parameters to be examined.

In this article, we describe shmoo plotting's very important role in testing to ensure that IC quality is maintained. We also explain how shmoo plotting can show that even digital ICs have analog dependencies.

Background

Developed in the early 1970s, shmoo plotting was specifically intended to examine problems with core memories. Core memories are a nonsemiconductor memory based on magnetic rings and were widely used in the 1960s and 1970s before LSI memories were available. Shmoo plotting likely took its name from either the appearance or the eternally obliging nature of Al Capp's cartoon figure Shmoo (see box) in test characterizations.

In its early days, shmoo plotting, coupled with the primitive form of ATE[1] available then, basically helped designers analyze problems only with power supply voltage or V_{DD}. Control of other parameters, such as timing and temperature, would await the arrival of commercial ATE in the mid-1970s, as Robert Huston told the authors in a 1996 personal communication.

0740-7475/97/$10.00 © 1997 IEEE

Originally, shmoo plots were created with ASCII character sets because the earliest computer printing devices, teletype machines, had very limited capability. (ASCII is still the preferred format for posting shmoo plots via e-mail.). Today, you can enjoy vastly improved ATE, displays, and printing, but the goal of shmoo plotting remains the same: to ensure the highest quality product that can be reliably manufactured with a given manufacturing process.[2] For safety-critical applications—such as automotive, avionics, medical, and other domains—shmoo plotting is an extremely important quality assurance tool.

Shmoo plotting as a test and characterization technique for digital devices has always been closely linked to the development of MOS IC technology. The technical literature, however, virtually ignores this basic technique of modern IC product engineering. Shmoo plotting is an art, as we explain in this article. The trick is to understand the fundamental design problems in MOS technology and to create tests for those critical conditions that condense a great deal of information into simple plots. Only then can designers and test engineers work together to improve product quality.

A basic shmoo tenet is that CMOS ICs are made of MOSFETs and parasitic capacitors. On the basis of fundamental equations that make shmoo plotting predictable, you can identify the capabilities of the MOSFET charging the parasitic load capacitance over different voltages and temperature ranges. If shmoo plot analysis reveals behavior other than what can be normally expected from a MOSFET charging a capacitor, you need to characterize this and validate the data sheet specifications for all process conditions.

Shmoo plots in IC testing

Shmoo plotting, which has a specific place in IC testing and characterization, can be used for some products but not others. Shmoo plotting is not typically used in IC production testing. It is used to verify that the data sheet specifications are realistic and that only one or two ICs in a million won't comply. Shmoo plotting is extensively used with most high-performance microprocessors, DRAMs, many ASICs, and sometimes for entire systems.[3]

Defects versus parameter deviations. Shmoo plotting can examine the design as a whole against its data sheet specifications by checking a limited number of prototype parts and projecting possible production process variations. However, if a specific IC has a minor soft defect, not a hard functional fault, then shmoo plotting is a tool that reveals information on that device alone.

Maly and colleagues[4] analyzed IC testing in 1986 and determined that testing has different stages, each with specific goals to prevent yield loss and to control the process to improve yield over the product lifetime. Shmoo plotting allows the sensitivities of the IC to process variations to be examined. If the process is critical to part of the specification, then the yield can be improved by changing the specification. This includes reducing the temperature range, limiting the clock conditions, or setting hard limits on the power supply voltage variation.

Is this the "shmoo" in shmoo plotting?

Who was the Shmoo? It was a cute little white, squash-shaped cartoon character created by Al Capp in his comic strip "Li'l Abner," in 1948 (see Figure A). Shmoos bred like rabbits and could produce any object at the drop of a hat. Since they loved to please humans, they would willingly pump out milk, eggs, filet mignon, caviar, or anything else if requested. At first glance the Shmoo seemed to herald the arrival of Utopia. Unfortunately, a plethora of Shmoos meant that people quit their jobs, stopped paying taxes, and civilization as we know it began to degenerate quickly—or so Al Capp sought to demonstrate in his mildly didactic way. In other words, there is such a thing as too much of a good thing. The great funnies-reading public was not so sure. Shmoos were enormously popular and thousands of Shmoo products were bought and sold before the Shmoo fad ran its course. Al Capp supposedly became so sick of Shmoos that he killed them off and banished them from the strip. They popped up again for another short appearance in "Li'l Abner" ten years later, were killed off again and this time were gone for good.

—*Whole Earth Review*, No. 57, Winter 1987

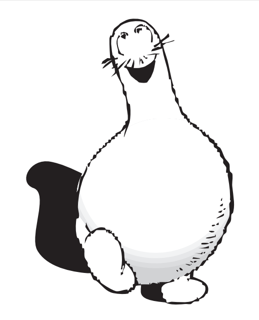


Figure A. *What a Shmoo looked like.*

Shmoo plotting of digital ICs is part of the early product characterization process where a product's performance is tested over the range of its specifications. The process establishes that the design is stable and can be manufactured with virtually zero yield loss, except for spot defects. In this sense, shmoo plotting verifies that the IC has been implemented to meet six-sigma design principles, or having 99.9997% accuracy.

Performance binning (physically separating devices by performance characteristics for later testing) is not performed for most ICs. As a result, testing all production parts for process variations can be avoided, which is cost-effective

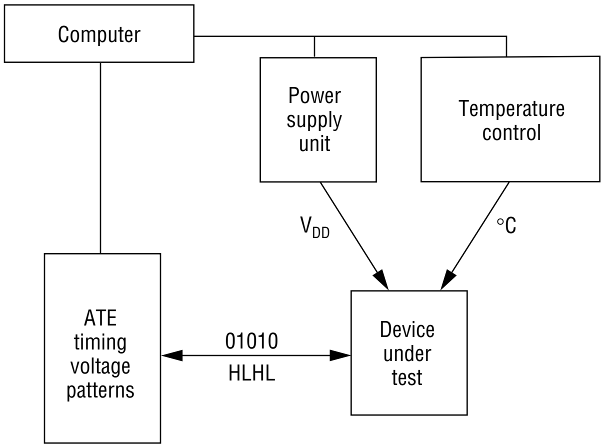


Figure 1. ATE setup for shmoo plotting. In digital testers, 01010 is for digital input signals to the device from the ATE; LHLHL is for digital output signals from the device to the ATE.

for high-volume production. Testing a device for process variations expressly implies functional testing. Because of the additional parameters to be tested, functional testing is far more time-consuming than simple defect-oriented tests, such as scan testing or I_{DDQ}.

Shmoo plotting a limited batch of ICs from production will not catch a hard defect in another batch of ICs, such as open or short conductor defects. Careful shmoo plotting, however, can determine the IC's performance variation and establish the need for process control. For example, shmoo plots might show that with MOSFET threshold voltages under 0.5 V, the IC won't work. By measuring the MOSFET threshold voltage directly during production, however, we can determine if the ICs in a particular batch will work or not.

Maly's proposed yield loss analysis method implies that for production testing there is a need to detect random spot defects in the process, necessary for all ICs. However, for performance issues, if process control is good, it is only necessary to plot a limited number of IC batches to maintain quality. With knowledge of the performance dependencies and different parameters gained from shmoo plotting, you can optimize the

- process
- design, such as clock systems, power supply, or critical paths
- wafer/batch monitoring parameters and procedures

You could also optimize the final test program with shmoo plotting results, although final test would normally already be optimized for defect detection to reduce test time.

Analog ICs. Shmoo plotting is rarely used with analog and mixed-signal ICs. This may seem strange because designers routinely use it on digital ICs to control the very parameter variations that cause such problems in analog design.

So why are analog ICs different? First, establishing pass/fail criteria for analog circuits is much more difficult than for digital. In fact, it is more important to establish the deviation from expected performance than a simple Boolean good or bad. Second, it requires much more effort to process the output data to determine if a limit has been exceeded. Testing analog devices requires digital signal processing, which is relatively slow. Third, analog designers try to avoid using the MOSFET parameters directly in circuit design because MOSFETs are sensitive to external parameters and vary on a batch-to-batch basis.

Analog designers generally use matching techniques to avoid dependence on the individual MOSFET characteristics, but the classical MOSFET characteristic is lost in shmoo plotting of analog circuits. This immediately negates the value of evaluating extrinsic parameters such as V_{DD} to assess performance. Designers should ensure that extrinsic parameters have little or no impact on mismatch. Any dependence should be device-specific.

MOSFET mismatches are strongly related to the spatial variation of parameters across a wafer during production, which is similar to spot defects, in a sense. Shmoo plotting of analog parts, then, would be useful only if wafer distribution information could be retained—that is, if you could test directly on the wafer. Even then the results would indicate only the grossest problems with the design or the process (such as a calibration of an ion implanter) in the production process.

Creating shmoo plots

Most modern ATE systems, some offering built-in data manipulation, let you create shmoo plots for IC testing. Figure 1 shows a typical ATE configuration.

An ATE system contains most of the hardware needed for shmoo plotting. An exception is the equipment for controlling device temperature, which must be added to the tester externally and is a slow process to control. Testing for temperature is difficult to perform at low temperatures because of condensation on the DUT boards and in the ATE itself in some cases.

In testing an IC's intrinsic performance, it's easier to change V_{DD} than temperature. For many cases, such as automotive applications, however, the range of temperatures an IC must endure has more impact on performance than the ±10% change from nominal V_{DD} found in most device specifications. Nevertheless, although validating temperature performance is important, it is, from a practical engineering viewpoint, one of the last issues to be fully verified during the characterization process.

Pass/fail algorithm. Graphical user interfaces (GUIs) simplify shmoo plotting somewhat. Behind this simplified interface will invariably be a procedure written in C or a similar language that implements the following algorithm:

```
start
  initialization: hardware, software,
  and IC for loop 1 to N (slow Y
  primary)
   intervention code for
   primary_y(parameter, tracking)
       for loop 1 to M (fast X primary)
           initialize IC
           intervention code for
           primary_x(parameter,
           tracking)
           (Pass or Fail) test (&vector,
           parameters)
           save test results
       end loop
  end loop
  store and display
end
```

For most ATE there's a simple GUI for design engineers to work on a specific device, and a complex one for test engineers who understand the complexity of the ATE. GUIs of course vary by vendor. In general, you fill in a simple form with five to ten fields, determine the two primary parameters to be varied, and assign an axis (*X* or *Y*). It's assumed in this algorithm that *X* is the fast primary parameter and *Y* the slow primary parameter, but this is arbitrary. Next, you assign ranges to the primary parameters, fix the other variables, and give a reference to the pattern sequencer for the selected test sequence. Optionally, you can provide titles and other cosmetic features.

Finally, you execute the test procedure, which will typically return a display such as that shown in Figure 2. Tradition dictates that the plot returns green for pass and red for failure.

Typically, the ATE's GUI interface lets you save the shmoo plot in a simple ASCII representation as an array, with strings giving the parameters ranges and values, and with various titles. No industry-standard format exists for saving shmoo plots, nor are translation tools provided, so it is difficult to move data from ATE to ATE. To transport, translate, and display shmoo plot data away from the originating tester, you typically need an ad hoc arrangement of Unix filters and small conversion utilities.

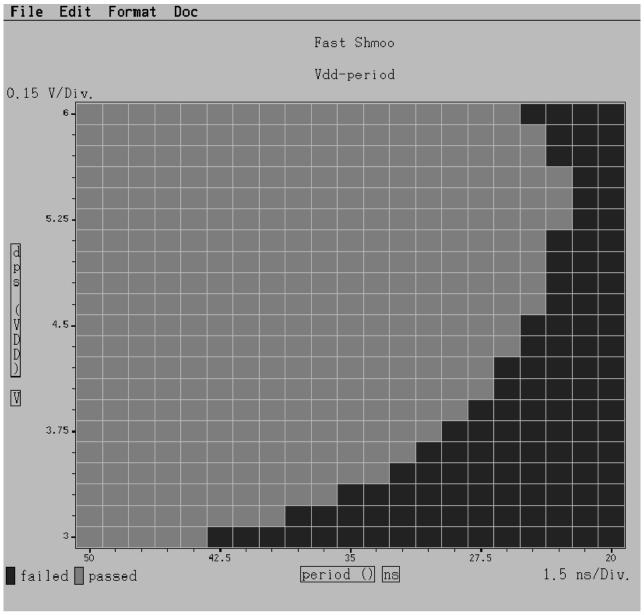


Figure 2. *Shmoo plot of V_{DD} versus period. (Note that this plot does not show the device performance that would normally be expected from a MOSFET. At 5.5 V, the performance peaks and then declines. Why? Further analysis reveals that the power supply is at fault: it had marginally exceeded its range but failed to warn the ATE of the problem. Shmoo plotting is a powerful technique, but it can have unexpected repercussions.)*

GUI complexities. Simple GUIs are not without limitations: some fixed parameters also relate to primary parameters that are changing during the plot. For example, the input levels and comparator thresholds depend on the selected V_{DD}. Also, various timing conditions of the data inputs can depend on the clocking signals. Tracking registers, sets of equations that describe how to change the secondary parameters as the primaries vary, handle such dependencies.

Another limitation is that sometimes much more complex operations must be performed during shmoo plotting. This operation can relate to the device, ATE, or external equipment (for example, a thermostream for device temperature control). Intervention code controls these actions. Usually, you write intervention code in C and attach it to the shmoo plot, although some ATE systems provide user-selectable intervention procedures or a small scripting language.

A third limitation is that, although this GUI interface is useful for interactive shmoo plotting, sampling many plots requires programming of the shmoo procedure. You can easily apply a basic shmoo plotting procedure in C to create special shmoo plots without using the GUI interface.

Fast-plot algorithms. The simple software algorithm is the simplest to create. Fast plots take less than ten seconds; acceptable plots take less than minute, while long plots take minutes. Faster methods for creating shmoo plots are possible, based on knowledge of their general form. Shmoo plots tend to have well-defined pass/fail regions. Edge-tracing algorithms can use this property to reduce the number of measurements. In the past, this type of algorithm was not always trustworthy because of the chance to miss a hole in a plot. This type of hole, or worm, is uncommon, however. If such holes do occur, they are more likely related to poor initialization sequences, ground bounce, or other issues concerning the DUT interface than actual device problems.

The art of shmoo plotting

The purpose of shmoo plotting is to provide useful information that, when applied, improves product quality in testing. It is an art for test engineers to gather the right kind of data for meaningful plotting and subsequent testing. The right kind of data requires that they understand IC design challenges in current technologies.

For the near term with deep-submicron circuits, the design engineer's greatest challenge is to match the IC's timing to the specification. Designs commonly use over 20% of the IC's area and 25% of the power simply to maintain a synchronous and deterministic operation. Larger ICs of course cost more money and ICs can waste energy in portable applications, using more batteries and affecting the environment in the longer term.

It's essential to understand how a circuit performs with respect to external signals. In terms of timing performance, for example, this includes clock generation and distribution (strings of CMOS inverters), both long and short critical paths, and flip-flop performance (close to meta-stable operation).

The great unknown in the early stage of IC characterization is the circuit's performance with respect to process variation. Therefore, it is important to establish the MOSFET capacitor-charging characteristic in shmoo plots. MOSFET characteristics enable performance prediction over the process spreads, but characteristics require more information about the IC's performance over wafers processed with different conditions or "process split batches." This is of course time-consuming and expensive. The MOSFET characteristic can be established in an IC through the extrinsic parameters, V_{DD} and temperature, to verify performance, as we explain next.

V_{DD} versus period. Using V_{DD} to control an IC's intrinsic performance is relatively quick and easy to plot. Plotting V_{DD} and timing results in the classical, most natural form of shmoo plot: the V_{DD} versus period, shown in Figure 2.

However, creating an accurate picture of IC performance for these two basic parameters can be difficult. Changing the period of IC timing also changes other timing conditions for the digital signals from the ATE. If you do not correctly track the other signals with period, then you can create a very false image of IC performance. Normally, a test engineer would work with either loose or tight timing conditions. In shmoo plots, the border between loose and tight timing may be unknown. To help in this regard, here is a procedure for a shmoo plot that will show V_{DD} versus period:

1. Determine the timing of all inputs and outputs with respect to the clock(s) from the specification.
2. Implement a robust test sequence to initialize the DUT.
3. Check the interface of the DUT with the ATE. Reflections on transmission lines can cause timing errors and must be eliminated by terminating the DUT outputs. Be aware that the MOSFET characteristics of the output drivers change with V_{DD} and temperature, so this is a compromise.
4. Check the device for ground bounce. This can be seen on the DUT's nonswitching outputs as major glitches using a good high-bandwidth scope.
5. Verify all setup and hold times for inputs and propagation delays for outputs as defined in the specification, for the given clock. (Learn how the device really works.)
6. Sample the outputs before the active clock edge. Adapt the test vectors, if necessary, to make the functional test pass. This avoids fails in the final shmoo plot caused by propagation delays at the outputs at shorter cycle periods.
7. Define relaxed duty cycle timing for all clocks related to the shmoo plot. This can be found and verified using the leading-trailing edge shmoo plot, as we explain later.
8. Program all the setup and hold times as an offset in the tracking registers of the shmoo plot. Use the specification for this if the pass region is to represent the actual working area of the specification.
9. Program the input and output levels according to the specification. Again, use the specification limits to accurately determine the pass region.
10. Run the V_{DD} versus period shmoo plot.

This procedure can be adapted for other shmoo plot types, but some care must be taken to avoid pitfalls. The parameters' influence on each other may require other shmoo plots to be examined before the final series of plots can be accurately made.

Other forms of shmoo plots. The original cartoon character Shmoo could give you anything you desired. Similarly, shmoo plotting can tell you virtually anything you want to know about a digital IC's performance. Of course, unique products need a unique, appropriate shmoo plot for analysis; these can be a characterization challenge. Shmoo plots are suitable where it is easy and quickly possible to control, with software, the primary plot parameters. This is more difficult if mechanical control is needed. For example, shmoo plotting an IC for resistance to alpha particles is time-consuming and hard to do.

Following are two examples of plots.

Leading-trailing edge shmoo plot. Sometimes it's neces-

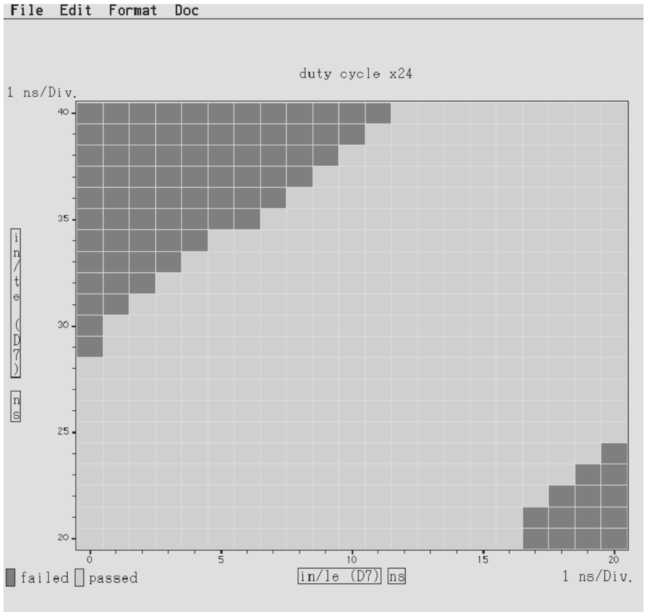


Figure 3. *Shmoo plot of the clock's trailing edge versus its leading edge, with V_{DD} and period as fixed parameters. This plot indicates device sensitivity to changes in duty cycle. The clock system's quality can be examined when this sort of plot is stacked against different V_{DD}.*

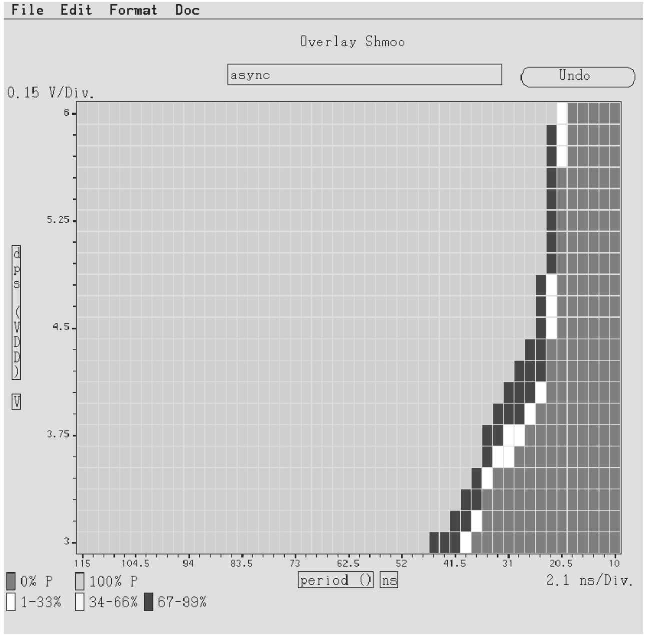


Figure 4. *A stacked shmoo plot of V_{DD} versus period for four different temperatures. DPS stands for device power supply. The percentages shown are a display artifact produced by the ATE.*

sary to use shmoo plots that don't influence MOSFET characteristics. You might need these, for example, to see if the IC performance is stable over a given range of timing conditions. This happens in Step 7 of the procedure just described. You would thus create a leading-trailing edge shmoo plot as shown in Figure 3. In this plot, the V_{DD} and period are fixed but the trailing and leading edges are used as primary parameters. This plot gives information on the sensitivity of the device to duty cycle changes. The IC works only when the duty cycle of the clock is well balanced. Either extremely short high or low periods result in functional failures. This type of plot can be stacked against different V_{DD} to examine the overall quality of the clock system.

Temperature shmoo plots. Because temperature control of an IC is a slow process, it is unusual to directly create a shmoo plot with temperature as a primary parameter. Instead, you might select two useful primary variables and create a set of shmoo plots at different temperatures. You can stack these plots with different colors or gray scales to histogram the number of pass/fails, as shown in Figure 4. In this case, four stacked temperature values create a histogrammed shmoo plot of V_{DD} and period.

Effective shmoo plots

Modern CMOS ICs differ from PMOS ICs only in the scale of integration, so essential MOSFET limitations should be the underlying characteristic you see when examining IC performance against various parameters. Other parts of the IC—the interconnect in polysilicon and metal—are mostly independent of the controllable parameters, both intrinsic process or extrinsic (V_{DD}, temperature, and timing). Moreover, you can easily monitor what dependence exists during the manufacturing process or measure it for the batch or wafer.

Technology developments have greatly improved the performance of the basic MOSFET since the 1970s, and design styles—specifically CMOS—have lessened circuits' performance dependence on individual transistors. The extrinsic parameters affecting IC performance have changed dramatically. Modern ICs are expected to work over a very wide temperature range; for example, in automotive applications they're expected to withstand –40 to 120 degrees C. Such applications also demand complex timing, clocking systems, and clock distribution systems on the IC itself.

Under these conditions, a useful plot is V_{DD} versus duty cycle, as shown in Figure 5 (next page). In this case the period is fixed, but the clock trailing edge is changed and plotted against V_{DD}. Clock period is set relatively long, and the leading edge of the clock and data signals are conservatively timed. This means the left edge of the shmoo plot should be realistic for internal timing conditions (such as short-path effects or flip-flop issues). The device performance can be seen at the right edge. This shmoo plot, simple yet mean-

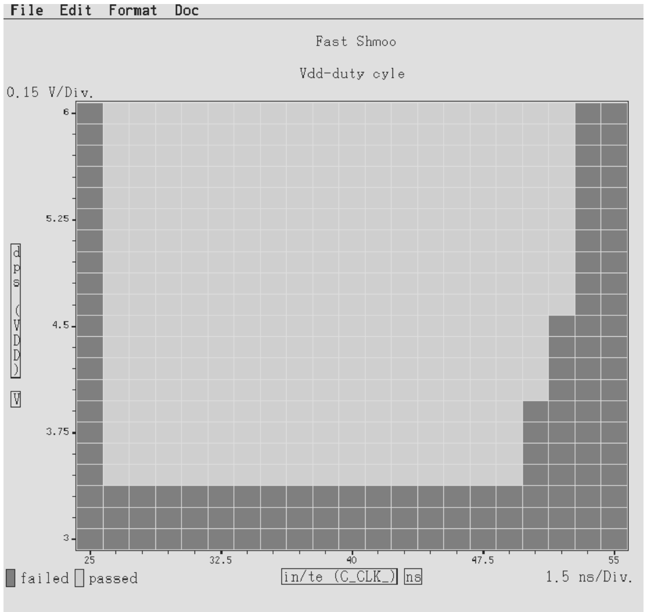


Figure 5. A shmoo plot of V_{DD} versus duty cycle. DPS stands for device power supply.

ingful, demands little of the tester timing performance because only one timing generator for the trailing edge of the clock changes. It's also less demanding because the period does not change directly.

Testability guidelines

Design rules for shmoo plotting and for testability are similar. However, "easily testable in test mode" may not be sufficient with shmoo plots. Customers use ICs in functional modes, not test modes. Thus we have the situation where shmoo plotting the IC in test mode is usually easier but is unfortunately not representative of normal modes of clocking or operation.

Test features in the IC's functional modes such as easy initialization, disable modes for PLLs, or control of multiple clocks are important for shmoo plotting. The four/five-pin IEEE Std. 1149.1 (JTAG) interface is a good method of controlling such features because the clocking is independent of functional clocks. This may not be true of IC-testable features.

Design issues crucial to shmoo plotting include the following:

- quickly initialized ICs (featuring self-test for RAMs and caches, or parallel load features for serial-loaded internal control memories)
- simple clocking (for example, avoiding large clock dividers; dividing 8 or greater; and carefully designing clock-edge adjustment schemes)
- power-on resets (disable)
- control of autonomous power-down features
- direct control of clock edges
- control of PLLs and FLLs (disable)
- direct control of digital interface of analog inputs

TECHNOLOGY TRENDS INDICATE that shmoo plotting will be a more important tool than in the past decade. These trends include low power supply voltage from 5 V to 3.3 or 2.5 V, active control of power supply to reduce dissipation, and challenges to the design tools such as clock tree synthesis. By not scaling V_{DD} with feature size, submicron technology had inherited the same excellent scaling properties as micron technology. This is changing as devices reach the limits of reliability in deep submicron because scaling requires V_{DD} to be continuously scaled in future generations. Low voltages imply that the intrinsic improvement in IC performance gained by MOSFET scaling without V_{DD} scaling will now need to be gained by better design.

In many manufacturing processes, V_{DD} is now being scaled close to the threshold voltage (V_T) of the micron scale CMOS technologies. The designer and process architect then must decide whether to scale V_T with V_{DD} (this will improve performance at the cost of background current due to subthreshold leakage) and negatively impact performance. In the latter case, this also brings the operating region of the MOSFET into the linear region, where the actual V_T of the individual device will be more critical. Because the MOSFET characteristic will once again play a larger role in the IC's overall performance, the expectation is that shmoo plotting will become increasingly important to maintain quality.

Despite increasing use, shmoo plotting will not necessarily be easier in the future. As IC performance improves, external timing effects start to dominate, so testing will require careful interface design to realistically picture device characteristics. An interesting problem is shmoo plotting bare die for applications in multichip modules: In many applications the system itself relies on the minimal parasitics of MCMs to gain performance. It is very difficult to build an interface with these characteristics on a modern tester. Shmoo plotting ICs for MCM applications is a challenge because the interface from IC to ATE is a limiting factor in performance. Accurately examining IC performance via this interface requires good models of the interaction of the interface, IC, and ATE.

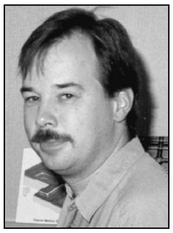
Acknowledgments

We thank Bob Huston and Fred Pool for providing unique insights into the earliest development and use of shmoo plotting in MOS IC testing. We also recognize Bob Huston's unique contribution, over 30 years in the industry, to shmoo plotting and to IC testing in general.

References

1. R.E. Huston, “Testing Semiconductor Memories,” *Proc. Int’l Test Conf.*, IEEE Computer Society, Los Alamitos, Calif., 1973, pp. 27-82.
2. R.E. Huston, *ITC Tutorial Notes*, IEEE Computer Society, Los Alamitos, Calif., 1996.
3. S.P. Allan, “Low-Cost Workstation with Enhanced Performance and I/O Capabilities,” *Hewlett-Packard J.*, June 1997, pp. 82-88.
4. W. Maly, A.J. Strojwas, and S.W. Director, “VLSI Yield Prediction and Estimation: A Unified Framework,” *IEEE Trans. Computer-Aided Design*, Vol. CAD-5, No. 1, Jan. 1986, pp. 114-130.

Keith Baker is the department head of Philips ED&T Test in Eindhoven, the Netherlands. His past research activities include both practical design verification and test strategies for digital and mixed-signal ICs. He was a founding member of the JTAG group in Europe in the 1980s, and a founder of QTAG at ITC 93 in Baltimore. Baker received a BSc and MSc in electrical and electronic engineering from the University of Essex, UK. He is a senior member of IEEE, a member of IEE, and a chartered engineer.



Jos van Beers is a VLSI test engineer at Philips Research Labs Eindhoven, the Netherlands. He has been involved in verification and characterization of various VLSI designs including EEPROMs, RAMs, digital, and mixed-signal designs. He received a BSc from Hogeschool Eindhoven.

Address questions or comments about this article to Keith Baker, ED&T, Philips Research Laboratories, Bldg. WAY 3 043, Prof. Holstlaan 4, 5656 AA Eindhoven, NL; baker@natlab.research.philips.com.